



Un micromonde pour aider les élèves à apprendre l'algèbre

Denis Bouhineau, Jean-François Nicaud, X. Pavard, Emmanuel Sander

► To cite this version:

Denis Bouhineau, Jean-François Nicaud, X. Pavard, Emmanuel Sander. Un micromonde pour aider les élèves à apprendre l'algèbre. *Sciences et Techniques Educatives*, 2001, 8 (1-2), pp.33–47. hal-00961980

HAL Id: hal-00961980

<https://hal.science/hal-00961980>

Submitted on 25 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Un micromonde pour aider les élèves à apprendre l'algèbre

Denis Bouhineau*, Jean-François Nicaud*, Xavier Pavard*, Emmanuel Sander**

*IRIN, 2 rue de la Houssinière, Université de Nantes, UFR Sciences
BP 92208, 44322 Nantes cedex 3, France

{bouhineau, nicaud, pavard}@irin.univ-nantes.fr

**ESA-CNRS "Cognition et Activités Finalisées", Université de Paris 8,
2, rue de la Liberté, 93526 Saint-Denis Cedex, France
sander@univ-paris8.fr

Résumé

Cet article décrit les principes de conception d'un micromonde dédié à la manipulation d'expressions algébriques. Ce micromonde contient un éditeur avancé avec des actions classiques et de la manipulation directe. La plupart des actions fonctionnent dans deux ou trois modes, les trois modes d'action étant : un mode "texte" qui manipule des caractères, un mode "structure" qui prend en compte la structure des expressions et un mode "équivalent" qui conserve l'équivalence des expressions. Le micromonde permet aussi de représenter des arbres de raisonnement. L'équivalence des expressions construites par l'élève est évaluée et l'élève est informé du résultat. L'article décrit aussi l'état actuel de la réalisation du micromonde dont le premier prototype a été opérationnel en février 2001.

Mots-clés: Algèbre, micromonde, manipulation directe.

1. Introduction

Des micromondes mathématiques ont été développés depuis la fin des années soixante. Le premier était Logo [Papert 80] consacré à la programmation récursive et à la géométrie. Les concepts de micromonde mathématique ont été définis dans la seconde moitié des années quatre-vingts [Thompson 87, Laborde 89, Balacheff et Sutherland 94] et les principales caractéristiques ont été dégagées : objets, relations, opérateurs et manipulation directe. Plusieurs micromondes ont été réalisés dans les domaines de la géométrie et de l'algèbre. Certains d'entre eux sont distribués comme des produits commerciaux et ont (ou ont eu) un succès réel, comme Logo et Cabri-Géomètre [Laborde 89]. Le dernier a été porté sur des calculatrices de poche et distribué comme logiciel par Texas Instruments. Les micromondes de géométrie fournissent une contribution réelle à l'étude du domaine et ont participé à la naissance d'un nouveau type de géométrie appelé géométrie dynamique.

Dans ce papier, nous considérons des micromondes pour l'algèbre. Algebraland [Foss 87] et le système de McArthur [McArthur et al. 87] peuvent être vus comme les premiers dans ce domaine. D'autres logiciels pour l'algèbre, comme ceux consacrés à la formalisation mathématique de problèmes énoncés en langue naturelle ('word problem' en anglais) [Koedinger et al. 97], ne sont pas concernés ici. Dans nos travaux précédents, nous avons réalisé ce que nous voulons considérer comme des micromondes *d'ordre 2* pour l'algèbre, nos systèmes étaient des prototypes avancés [Nicaud et al. 90, 94]. Nous avons conduit de nombreuses expériences, analysé le comportement des élèves et les modes d'apprentissages permis par ces systèmes [Nguyen-Xuan et al. 93, 99]. Nous concevons maintenant et réalisons ce que nous voulons considérer comme un micromonde *d'ordre 1* pour l'algèbre. Ce système vise à devenir un produit commercial. Dans l'avenir, nous combinerons les particularités des micromondes d'ordre 1 et 2.

La question fondamentale est : Que peut être un micromonde pour l'algèbre ? Dans ce papier, nous définissons les micromondes d'ordre 1 et 2. Ensuite, nous détaillons les micromondes d'ordre 1 et décrivons les principales caractéristiques du système que nous développons ainsi que l'état actuel de sa réalisation.

2. Résumé de l'algèbre

Avant d'essayer de répondre à la question : *Qu'est-ce qu'un micromonde pour l'algèbre*, nous devons résumer le domaine et préciser les objets et les relations entre ces objets.

Le domaine de l'algèbre que nous considérons est composé d'expressions algébriques et de règles de transformation de ces expressions. Il y a des types de problème comme la factorisation, la résolution d'équations et le calcul de dérivées. La résolution d'un problème s'effectue par transformation de l'expression du problème donné à l'aide des règles de transformation correctes jusqu'à ce qu'une forme résolue soit atteinte. Les règles de transformation correctes sont les règles qui maintiennent l'équivalence des expressions.

Les objets concrets de l'algèbre sont des expressions. Une expression est une succession de chiffres (comme 12), un symbole (comme une inconnue x , un paramètre m , une constante π) ou un opérateur appliqué à des expressions. Une représentation à deux dimensions internationale, que nous appelons la *représentation habituelle*, a été adoptée pour les expressions. Par exemple, $\frac{\sqrt{x}}{12+x^2}$ est un quotient. Un

quotient est l'application de l'opérateur de *division* à deux arguments. Ici, les arguments sont \sqrt{x} et $12+x^2$. Le premier argument est la racine carrée de x (l'opérateur *racine carré* appliqué à la variable x) ; le deuxième argument est la somme de 12 et x^2 , etc. Plusieurs règles précisent ce que sont les *expressions bien formées*, en particulier la règle d'arité (par exemple, "=" a deux arguments), les règles de type (par exemple, "=" s'applique aux arguments algébriques et fournit un résultat booléen (ainsi $a=b$ ne peut être l'argument d'une somme parce qu'une somme s'applique à des arguments algébriques)). Nous notons \mathcal{E} l'ensemble des expressions *bien formées* pour un domaine de problème donné.

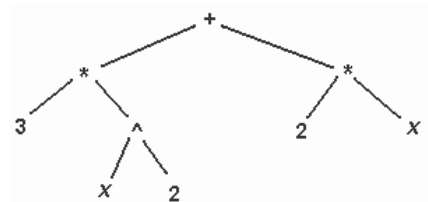


Figure 1: Représentation sous forme d'arbre de $3x^2+2x$

La représentation habituelle n'est pas la plus simple. La plus simple est une représentation sous forme d'arbre (voir la figure 1) dans laquelle les opérateurs et leurs arguments sont très visibles et dans laquelle il est évident de trouver à quels arguments un opérateur s'applique. Mais une représentation sous forme d'arbre est lourde. Cela explique pourquoi la représentation habituelle a été construite. Les principales difficultés dans la représentation habituelle viennent d'opérateurs cachés (comme la *multiplication* dans $2x$) et de la priorité d'opérateurs ($x+2=0$ est l'opérateur $=$ appliquée à $x+2$ et 0 parce que la priorité de $+$ est plus haute que la priorité de $=$). Les parenthèses sont parfois nécessaires, comme dans $x(x+1)$; elles ne le sont pas dans la représentation d'arbre.

Les objets abstraits de base de l'algèbre sont les nombres. Nous notons \mathcal{K} l'ensemble des nombres considérés pour un domaine de problème donné. Dans l'enseignement secondaire, c'est généralement l'ensemble de nombre le plus grand qui a été présenté et, pendant la plus longue période, ce sont les réels. Il y a un morphisme fondamental $\mathcal{M}1$ de \mathcal{E} vers \mathcal{K} qui associe les opérateurs aux fonctions (par exemple, $+$ est associé à l'addition des nombres) et qui associe n'importe quel couple constitué d'une expression et une substitution à nombre (par exemple, $2x+6$ avec $[x : 5]$ est associé à 16). Ce morphisme permet d'employer des opérateurs dans la représentation de nombres. Les entiers naturels ont des représentations canoniques dans la base 10 faites de successions de chiffres. D'autres nombres ont des représentations qui emploient des opérateurs et des chiffres (comme -12 et $\frac{4}{3}$). Certains nombres ne peuvent pas être représentés avec des opérateurs et des chiffres et sont représentés avec des symboles (comme π).

Comme il est dit ci-dessus, les expressions sont transformées pour résoudre des problèmes. Par exemple, $\langle \text{factoriser } (x+3)^2-1 \rangle$ peut être résolu en appliquant la règle $A^2-B^2 \rightarrow (A-B)(A+B)$ à $(x+3)^2-1$, ce qui produit $[(x+3)-1][(x+3)+1]$, en supprimant ensuite les parenthèses autour $x+3$, on obtient $(x+3-1)(x+3+1)$, en réduisant finalement les nombres, ce qui donne $(x+2)(x+4)$. Les règles de transformation sont des objets du domaine. L'application de règles correctes n'est pas la seule façon de résoudre des problèmes. Par exemple, l'élève peut factoriser x^2-5x+6 en donnant directement la solution $(x-3)(x-2)$ parce qu'il a deviné les racines du polynôme. En réalité il y a un domaine fondamental qui donne une signification aux expressions. Le dernier exemple peut être justifié par "les expressions x^2-5x+6 et $(x-3)(x-2)$ sont équivalentes parce qu'elles sont associées au même polynôme". Pour des expressions ayant un type algébrique, le domaine fondamental le plus souvent employé est l'ensemble des fonctions de \mathcal{K} dans \mathcal{K} mais ce peut aussi être les polynômes ou les fractions rationnelles sur \mathcal{K} . Pour des expressions ayant un type booléen (comme les équations), le domaine fondamental est l'ensemble des fonctions de \mathcal{K} dans $\{\text{faux}, \text{vrai}\}$.

Notons \mathcal{F} le domaine fondamental et appelons fonctions les éléments de \mathcal{F} , même quand ce sont des polynômes ou des fractions rationnelles. Il y a un morphisme fondamental \mathcal{M}_2 de \mathcal{E} dans \mathcal{F} qui associe les opérateurs aux fonctions de fonctions (par exemple, $+$ est associé à l'addition des fonctions) et qui associe n'importe quelle expression à une fonction (par exemple, $2x+6$ est associée à la fonction $x \rightarrow 2x+6$). Les fonctions sont les objets abstraits fondamentaux parce qu'ils fournissent l'équivalence des expressions : deux expressions sont équivalentes si et seulement si elles sont associées à la même fonction de \mathcal{F} par le morphisme \mathcal{M}_2 . Cela inclut l'équivalence pour le type algébrique (x^2-5x+6 et $(x-3)(x-2)$ sont équivalents) et pour le type logique ($x=1$ est équivalent à $(x^2+1)(x-1)=0$ quand \mathcal{K} est l'ensemble des réels).

3. Que peut être un micromonde pour l'algèbre ?

Selon Laborde [89], un micromonde est un monde d'objets et de relations. Il y a un jeu d'opérateurs capables d'agir sur les objets, de créer de nouveaux objets avec de nouvelles relations. La manipulation directe a une place importante dans les micromondes. Balacheff et Sutherland [94] ont mis en lumière *un domaine épistémologique de validité* liant les représentations et ce qu'elles sont censées représenter. Thompson [87] dit que les micromondes mathématiques doivent faciliter le processus de construction des objets et des relations et permettre de se concentrer sur la construction du sens. Ils doivent employer un modèle non incorporé dans un programme d'études et ils ne doivent pas enseigner.

En algèbre formelle, comme indiqué ci-dessus, les objets concrets sont des expressions. La relation de base entre les objets est la relation structurelle sur les expressions : 12 est le premier argument du deuxième argument de $\frac{\sqrt{x}}{12+x^2}$. La deuxième relation est la relation d'équivalence. Il y a deux types principaux d'opérateurs.

Les opérateurs du premier type permettent de construire n'importe quelle expression bien formée : ce sont les opérateurs des *éditeurs d'expressions algébriques*. Nous nommons *ordre 1* les micromondes algébriques qui se concentrent sur ces opérateurs. Des micromondes algébriques d'ordre 1 sont inclus dans beaucoup de systèmes de traitement de texte. Ils sont appelés *éditeurs d'équation*. Nous montrerons ci-dessous que la plupart d'entre eux ont un rapport pauvre avec les objets sémantiques de l'algèbre. Le système de McArthur [McArthur et al. 87] a certaines des caractéristiques des micromondes algébriques d'ordre 1.

Les opérateurs du second type sont les règles de transformation qui permettent de construire de nouveaux pas dans un processus de raisonnement. Nous nommons *ordre 2* les micromondes algébriques qui se concentrent sur des règles de transformation. Plusieurs micromondes algébriques d'ordre 2 ont été réalisés dans le passé [Bundy et Welham 81, Foss 87, Thompson 89, Olivier et Zukerman 90, Beeson 96]. Les systèmes que nous avons développés précédemment dans notre équipe étaient de ce type. Les étudiants résolvaient des problèmes en appliquant des règles de transformation (choisies dans un menu) à des sous-expressions sélectionnées avec la souris. Les étudiants ne pouvaient pas construire un pas de calcul par une autre voie. La communauté n'a pas vraiment considéré ces systèmes comme des micromondes, peut-être

parce qu'ils enseignent (chose refusée par Thompson), peut-être parce qu'ils n'ont pas les opérateurs d'ordre 1, peut-être parce que les personnes qui ont réalisé ces systèmes ont plus insisté sur les aspects de tuteur.

Dans le reste du papier, nous nous concentrons sur les micromondes algébriques d'ordre 1 ; nous les appelons juste micromondes algébriques. Dans la section 4., nous considérons les caractéristiques classiques d'un éditeur pour un micromonde algébrique. Dans la section 5., nous abordons la question de la manipulation directe.

4. Édition classique pour un micromonde algébrique

Objets et relations structurelles

Un micromonde doit permettre à l'utilisateur de créer des objets. Dans le cas de l'algèbre, cela signifie contenir un éditeur pour les expressions algébriques. La fidélité visuelle impose la représentation 2D habituelle pour l'interface de l'éditeur. Cette représentation contient des parties de texte, que l'on peut voir comme des successions de caractères (comme $2x+1$), et des blocs bien délimités (comme $\frac{x}{1+x}$ et $\sqrt{x+1}$).

La fidélité conceptuelle impose de prendre en considération les relations structurelles. Nous considérons qu'un premier pas dans cette direction consiste à avoir en permanence une expression ayant une structure et le plus souvent possible une expression formée. Pour cela, des arguments manquants sont remplacés par quelque chose, comme "?" (par exemple, taper + après $2x$ donne $2x+?$).

Les actions principales d'un éditeur sont : l'entrée, la suppression, la sélection, l'application d'un opérateur à quelque chose, la copie de quelque chose, la suppression de quelque chose, la coupe de quelque chose, le collage de quelque chose. Considérons ces actions du point de vue de la relation structurelle. Cela implique que le susdit « quelque chose » est une sous-expression. Nous appelons *mode structure*, un mode où des sous-expressions bien formées sont conservées pendant les actions ; nous appelons *mode texte*, un mode où des sous-expressions bien formées ne sont pas conservées comme sous-expressions bien-formées pendant l'action. Dans l'interaction avec l'ordinateur, l'utilisateur peut choisir le mode. Noter que les deux modes sont utiles, le premier pour éviter de détruire ce qui est déjà construit, le second pour permettre à l'utilisateur de faire tout ce qu'il veut.

Propositions pour la mise en oeuvre des actions

Entrée à partir du clavier (ou entrée en cliquant sur un bouton), quand l'expression a un point d'insertion : pour obtenir une expression bien formée, l'entrée d'un opérateur de texte ajoute "?" quand cela est nécessaire (par exemple, taper + après $2x$ donne $2x+?$) ; dans le mode structure, l'entrée d'un opérateur de bloc ajoute "?" pour chaque argument (par exemple, taper / après x dans xy donne $x\frac{?}{?}y$) ; dans le mode texte, l'entrée d'un opérateur de bloc utilise l'environnement pour les arguments (par exemple, taper / après x dans xy donne $\frac{x}{y}$) ; dans le mode structure, l'entrée d'une parenthèse ouvrante transforme le pointeur de souris en une parenthèse fermante « à déposer immédiatement » ; dans le mode texte, l'entrée d'une parenthèse ouvrante place cette parenthèse seule, en lui donnant la couleur bleue signifiant « expression mal formée ».

Supprimer à droite ou à gauche du curseur d'insertion : dans une partie de texte, un caractère seulement est supprimé (par exemple, pour $2x+1$, le curseur étant entre 2 et x , < del > donne $2+1$) sauf quand le caractère est "?" l'opérateur l'exigeant étant aussi supprimé (par exemple, pour $2+?+1$, le curseur étant entre + et ?, < del > donne $2+1$) ; quand un bloc est concerné (incluant des parenthèses), le bloc entier est supprimé dans le mode structure, l'opérateur seulement est supprimé dans le mode texte (par exemple, pour $2\sqrt{x}$, le curseur étant après 2, < del > donne 2 dans le mode structure et $2x$ dans le mode de texte).

Sélectionner une sous-expression : la relation structurelle suggère de sélectionner seulement des sous-expressions bien formées. Noter qu'une sous-expression bien formée peut avoir plusieurs composantes comme $2x+4$ dans $2x+y+4$. L'idée naturelle, pour le choix d'une sous-expression, est de considérer que ce qui est sélectionné est la sous-expression la plus petite qui inclut la zone balayée. Ces actions de sélection ont été mises en oeuvre dans nos prototypes précédents et ont été appréciées par les élèves et les enseignants.

Appliquer un opérateur : quand une sous-expression est sélectionnée, nous considérons que l'entrée d'un opérateur unaire signifie "appliquer cet opérateur à la sous-expression". Certains opérateurs binaires peuvent aussi être appliqués sous des formes introduisant des "?" (par exemple : mettre en numérateur d'une fraction).

Copier une sous-expression : copie la sous-expression choisie dans le presse-papiers.

Supprimer une sous-expression : supprime la sous-expression choisie et un opérateur quand nécessaire.

Couper une sous-expression : c'est la copie d'une sous-expression + la suppression d'une sous-expression.

Collez une sous-expression : la sous-expression est insérée à la place du curseur ou de la sélection. Dans le mode structure, des parenthèses sont ajoutées quand nécessaire (coller $x+1$ après 2 dans $2y$ donne $2(x+1)y$). Dans le mode texte, les parenthèses ne sont pas ajoutées (coller $x+1$ après 2 dans $2y$ donne $2x+1y$).

Raisonnement et la relation d'équivalence

Une activité importante de l'algèbre formelle consiste à résoudre des problèmes par des transformations successives d'expressions, en conservant l'équivalence. Après l'édition d'expressions nous devons considérer la représentation du processus de raisonnement dans un micromonde algébrique. Un processus de raisonnement comporte non seulement des transformations successives d'expressions, avec la conservation de l'équivalence, mais aussi des retours en arrière et des sous-problèmes. Un retour en arrière est parfois réalisé lors de la résolution d'un problème. Il consiste à retourner à une ancienne étape du raisonnement pour essayer une autre façon de résoudre le problème (donc l'ensemble des étapes a la forme d'un arbre). Un sous-problème est parfois extrait et résolu pour aider à résoudre le problème. Étant donné un problème $\langle T \ e \rangle$, T étant le type et e l'expression, la situation la plus fréquente pour un sous-problème consiste à avoir un problème $\langle T1 \ e1 \rangle$, où $e1$ est une sous-expression de e , telle que la forme résolue du sous-problème puisse profiter au problème original. Par exemple, $\langle \text{résoudre } A=0 \rangle$, quand A est un polynôme de degré supérieur à 1, tire profit de la résolution du sous-problème $\langle \text{factoriser } A \rangle$.

Nous considérons qu'un micromonde algébrique doit représenter le processus de raisonnement, l'équivalence entre les expressions, les retours en arrière et les sous-problèmes.

Propositions pour la mise en oeuvre des processus de raisonnement

Dans nos prototypes précédents, nous avons mis en oeuvre toutes les caractéristiques précédentes. La liaison entre des pas équivalents était une flèche (la signification étant "ce pas est produit à partir de ce pas"). Nous pensons maintenant qu'il serait meilleur d'employer une flèche double signifiant "ces expressions sont équivalentes". Les retours en arrière ont été mis en oeuvre par la possibilité d'avoir plusieurs successeurs pour n'importe quel pas. Nous pensons bon de maintenir cela. Les sous-problèmes ont été liés aux problèmes initiaux avec une flèche spéciale. Nous pensons maintenant qu'il serait préférable de résoudre les sous-problèmes sur d'autres endroits de la feuille ou sur d'autres feuilles.

Systèmes existants

Les éditeurs d'équations de Microsoft, FrameMaker, StarOffice [Microsoft Equation, Adobe FrameMaker, StarOffice] offrent à l'utilisateur une interface spécifique pour définir et représenter des expressions algébriques. La disposition des éléments d'une expression dans les deux dimensions du plan en respectant la représentation habituelle et les normes internationales, le dessin spécifique de nombreux opérateurs algébriques, l'adoption de certaines conventions typographiques pour l'écriture de nombres, des variables et des noms de fonction, justifient une interface indépendante. Les formules et les expressions sont obtenues à l'aide de juxtapositions prédéfinies de boîtes à remplir de texte ou d'autres formules. Le résultat obtenu est généralement satisfaisant sur le plan graphique mais certaines expressions sont très mal représentées et l'aspect algébrique des éditeurs est souvent beaucoup moins développé que leurs fonctionnalités graphiques. Par exemple, il est commun de ne pas être capable de différencier les expressions du type de 2^{22} – c'est à dire $(2^2)^2$ – de l'expression 2^{22} – c'est à dire 2 à la puissance 22 – (Adobe FrameMaker, StarOffice). L'aspect graphique est souvent très développé sans qu'un contrôle mathématique

ne soit effectué, ainsi, l'utilisateur peut déplacer les éléments de son choix à l'intérieur d'une expression pour ajuster leurs positions, au pixel près ; plusieurs types de parenthèses sont prévus pour obtenir $((((x+1)x+2)x+3)x+4)$ ou $((((x+1)x+2)x+3)x+4)$; les tailles, styles et polices des caractères sont ajustables par l'utilisateur pour un meilleur rendu visuel, mais tout cela est exécuté sans aucun contrôle algébrique. Ainsi certains objets obtenus sont algébriquement mal formés, par exemple, des expressions incomplètes, des parenthèses non équilibrées (Microsoft Equation). Parfois, en plus, les constructions par juxtaposition de boîtes respectent les structures algébriques sous-jacentes mais le résultat est négatif. Par exemple, dans $2x^3$, il est impossible de sélectionner x^3 qui se trouve dans deux boîtes différentes, au contraire $x+2$ peut être sélectionné dans $5x+23$, bien que ce ne soit pas une sous-expression algébrique, parce que les éléments ne sont pas dans des boîtes différentes. D'un point de vue général, toutes les manipulations permises sur l'expression doivent être exécutées en respectant la structure de boîtes et cette structure est plus graphique qu'algébrique, comme conséquence, déplacer et interpréter la place du curseur est une opération difficile parce qu'il y a beaucoup de positions graphiques qui correspondent à une même position algébrique.

5. Manipulation directe

Dans un micromonde, la manipulation directe permet de déplacer des objets en prenant en compte les relations structurelles. Les expressions, sous-expressions et opérateurs peuvent être manipulés selon leurs relations structurelles, en conservant l'équivalence ou sans s'occuper des relations structurelles et de l'équivalence.

Glisser-déposer d'une sous-expression

On peut faire glisser une sous-expression choisie s et la déposer à l'intérieur d'une expression e comme un mot dans un traitement de texte. Une telle action change l'expression. Soit u l'expression obtenue, dans le mode structure, les contraintes que nous considérons viennent des relations structurelles, ce sont : (1) u est une expression bien formée, (2) s est une sous-expression de u , (3) les sous-expressions de e qui n'incluent pas s sont laissées inchangées. Noter que ces contraintes introduisent parfois des parenthèses autour de s . Dans le mode texte, la seule contrainte est la première et les parenthèses ne sont jamais ajoutées. Le but du glisser-déposer dans ces deux modes est de permettre à l'utilisateur de construire l'expression qu'il veut.

Le but du glisser-déposer dans le mode équivalent est très différent ; il doit conserver l'équivalence. Donc nous ajoutons la contrainte (4) u doit être équivalent à e . Avec les contraintes (1) à (4), le seul mouvement général qui peut être effectué consiste en déplacer un argument d'un opérateur commutatif. Pour obtenir un glisser-déposer dans le mode équivalent plus puissant, nous retirons la contrainte (3) de telle sorte que les sous-expressions de e qui n'incluent pas s puissent être changées.

Prenons comme un exemple l'expression $(y-1)(x+x^2)$, sélectionnons la première occurrence de x et glissons la entre les deux parenthèses. Nous pouvons nous attendre à obtenir $(y-1)x(1+x)$ qui est une factorisation. Une explication de la factorisation dans un tel contexte peut être que l'expression déplacée est un argument d'une somme, qui est un argument d'un produit, l'action étant exécutée comme "prendre x comme un facteur commun de la somme et l'insérer comme un facteur du produit". Noter qu'il est nécessaire de retirer la contrainte (3) pour cela, car x^2 a été changé en x . La table 1 fournit d'autres exemples pour les opérateurs de base.

#	Expression	Sous-expression sélectionnée	Endroit du déposer	Résultat attendu
1	$3x-1+x^2$	x^2	avant 3	x^2+3x-1
2	$(y-1)(x+x^2)$	première occurrence de x	entre)($(y-1)x(1+x)$
3	$(y-1)(x+2)$	x	entre)($(y-1)x(1+\frac{2}{x})$
4	$(xy)^2$	x	avant (x^2y^2
5	$(x+x^2)^2$	première occurrence de x	avant ($x^2(1+x)^2$

6	$\sqrt{4+x}$	4	avant la racine	$2\sqrt{1+\frac{x}{4}}$
7	$\sqrt{2+x}$	2	avant la racine	$\sqrt{2}\sqrt{1+\frac{x}{2}}$
8	$\frac{x}{x+y}$	première occurrence de x	dénominateur	$\frac{1}{1+\frac{y}{x}}$
9	$\frac{x+x^2}{x^2+x^3}$	première occurrence de x	dénominateur	$\frac{1+x}{x+x^2}$
10	$3=x$	x	avant 3	$x=3$
11	$3x=12$	3	avant 1	$x=4$

Table 1, exemples de glisser-déposer dans le mode équivalent pour les opérateurs de base.

L'examen de ces exemples montre que l'on donne une place principale au concept de facteur. Même dans des situations où aucun produit n'apparaît, les facteurs sont employés (par exemple, #5 et #6). Ces exemples suggèrent de considérer que le glisser-déposer en mode équivalent comme "l'extraction ou l'insertion d'une sous-expression comme un facteur quand c'est possible" (ce qui diffère fortement de l'explication suggérée dans le paragraphe précédent). Nous devons faire remarquer que les résultats sont construits par l'application de règles algébriques (ce qui explique que l'équivalence soit conservée) et que des règles de réduction sont parfois appliquées (quand 2 est inséré dans $3x-6$, nous nous attendons à trouver $6x-12$) ce qui peut faire disparaître la sous-expression (dans ce cas, l'inversion du processus n'est pas possible).

En considérant la structure, il y a plusieurs situations générales pour un glisser-déposé : (1) un argument d'un opérateur est déposé à la place d'un autre argument (par exemple, #1, #8, #10, inverse de #2, #3, #5, #6, #7); (2) une sous-expression est extraite (par exemple, #2, #3, #4, #5, #6, #7); (3) une sous-expression est extraite puis introduite dans une autre (par exemple, inverse de #2, #3, #4, #5, #6, #7). Cela concerne le mode structure comme le mode équivalent.

Il est naturel de considérer la position de la souris comme la destination pour le déposer. Mais cette position ne détermine pas une position unique dans la structure, par exemple, la position après = dans $3x=x^2(x-1)+2$ peut être considérée comme une position dans la somme, dans le produit ou dans la puissance. Ce problème concerne le mode structuré comme le mode équivalent.

Pour les opérateurs qui ne sont pas de base, le glisser-déposer en mode équivalent peut être défini pour quelques situations comme dans la table 2. Noter que les actions sont complexes et que la relation avec les facteurs disparaît.

Expression	Sous-expression sélectionnée	Position du déposé	Résultat attendu
$\ln(xy)$	x	avant \ln	$\ln x + \ln y$
$\ln x^3$	3	avant \ln	$3 \ln x$
$\cos(x+y)$	x	avant \cos	$\cos x \cos y - \sin x \sin y$

Table 2, exemples de glisser-déposer dans le mode équivalent pour les autres opérateurs.

Manipulation d'opérateurs

Les parenthèses et les opérateurs de bloc définissent des espaces que l'utilisateur peut vouloir changer. Pour cela, nous considérons les manipulations directes de ces opérateurs : "(" et ")" peuvent être déplacés indépendamment par glisser-déposer ; le champ des barres de fraction et des radicaux peut être changé avec des poignées qui apparaissent quand l'opérateur est choisi. Par exemple, $2x + \sqrt{3y+2} + 5$ peut être transformé $2x + \sqrt{3y+2+5}$ ou en $2x + \sqrt{3y} + 2 + 5$ en utilisant la poignée de droite du signe radical.

Systèmes existants

Le calculateur graphique [Graphing Calculator] qui équipe les ordinateurs Apple depuis 1994 présente un manipulateur d'expressions algébriques très convaincant, bien qu'il soit limité aux fonctions usuelles. Les expressions sont toujours bien formées et non ambiguës. De plus, ce logiciel permet de manipuler les expressions algébriquement avec la souris. Selon son concepteur "le calculateur préserve l'égalité" pendant ces manipulations. Mais ce dernier point n'est pas toujours parfaitement respecté. Par exemple, en utilisant le glisser-déposer, $\sqrt{\frac{a}{b}}$ peut être transformée en $\frac{\sqrt{a}}{\sqrt{b}}$ et $x+1$ peut être transformée en $x(1+\frac{1}{x})$ sans l'indication d'un domaine de validité. Seuls les opérandes peuvent être manipulés avec ce logiciel. Nous aurions aimé pouvoir déplacer les parenthèses ou les signes « = », prolonger les barres de fraction ou les radicaux. Finalement, on peut regretter que l'application de ces manipulations complexifie graduellement les expressions et que ces manipulations ne soient pas réversibles : replacée dans sa position initiale, une expression peut ne pas retrouver sa forme initiale.

6. Le micromonde APLUSIX pour algèbre

Notre équipe a planifié la réalisation d'un micromonde algébrique d'ordre 1 en juin 2000. L'agenda comporte la réalisation d'un premier prototype pour janvier 2001 et une première expérience dans le début de 2001. Cet agenda est presque respecté (un mois de retard). Le développement est effectué avec Borland Delphi pour Windows. L'état actuel est décrit ci-dessous.

L'affichage 2D des expressions est effectué pour les opérateurs suivants : + - × / ^ √ sin cos tan cot = ≠ < > ≤ ≥ et ou non (voir la figure2).

La sélection, y compris la sélection de plusieurs arguments d'un opérateur commutatif, est mise en oeuvre (voir la figure 3).

Insertion, suppression, copie, coupe, collage sont mis en place dans les modes texte et structure. Les parenthèses non équilibrées (dessinées en bleu) sont considérées comme des opérateurs unaires particuliers afin de permettre de sélectionner, copier, couper, coller.

La manipulation directe est réalisée dans les modes texte et structure (figure 4). Le glisser-déposer en mode équivalent sera mise en oeuvre ultérieurement, juste pour déplacer un argument dans un opérateur commutatif. Un glisser-déposer plus complexe, en mode équivalent, n'est pas nécessaire pour nos premières expériences avec des élèves de 15 ans. Un glisser-déposer en mode équivalent plus complexe nécessite une réflexion plus profonde ; notre but n'est pas de mettre en oeuvre un joli mécanisme, mais un mécanisme ayant une valeur épistémologique.

La manipulation des opérateurs sera mise en oeuvre ultérieurement.

L'affichage de l'arbre du raisonnement est mise en place (figure 5).

L'évaluation de l'équivalence entre deux expressions A et B est réalisée quand A et B sont des polynômes d'une variable. L'équivalence de deux équations avec une inconnue et un degré inférieur à 5 sera mise en oeuvre prochainement (en résolvant les équations avec des formules : discriminant pour le degré 2, formules de Cardan pour le degré 3, formules de Ferrari pour le degré 4).

L'enregistrement des interactions dans des fichiers sera bientôt mis en oeuvre. Chaque action élémentaire sera enregistrée pour analyser le comportement des élèves en laboratoire, à la main ou avec des programmes. Nous installerons aussi, plus tard, une fonction « rejouer » permettant de voir toutes les actions des étudiants à l'écran.

$$\frac{\left(\frac{9x^2}{3} - \frac{2}{3}\right)^4}{7} + a^{m^2} + f^{2^3} = \sqrt{3x+5} - \sqrt{\sqrt{\frac{5}{2x^3 + 5^3}}}$$

Figure 2: Affichage d'une expression en représentation 2D usuelle

$$x^2 + (x-2)(x+3) = 4$$

Figure 3: Sélection de plusieurs arguments d'un opérateur commutatif

$$2y + 3(x^2) \quad 2 - x^2y + 3(x) \quad 2(-x^3)y + 3(x)$$

Figure 4: un glisser-déposer. L'expression sélectionnée (à gauche) est déposée entre 2 et y en mode texte au milieu et en mode structure à droite.

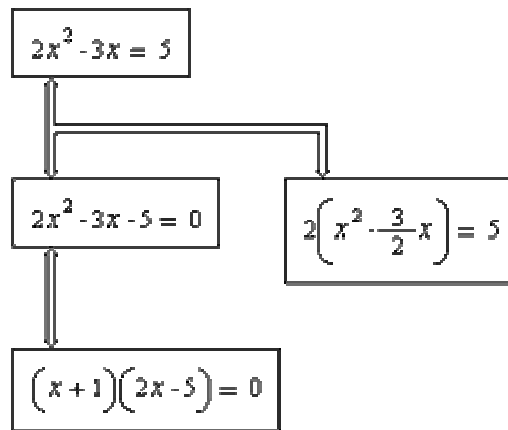


Figure 5: un arbre de raisonnement

Le micromonde Aplusix contient deux activités principales : d'abord permettre à l'étudiant de construire des expressions, ensuite permettre à l'étudiant de résoudre un problème portant sur une expression algébrique. Les premières expériences seront conduites avec des étudiants de 15 ans au début de 2001 pour évaluer ces deux activités, c'est-à-dire, pour évaluer l'éditeur et l'environnement de raisonnement. Comme l'éditeur n'est pas limité aux expressions simples, il sera aussi évalué avec des chercheurs qui manipulent habituellement des expressions algébriques.

Des réalisations ultérieures dans le cadre de notre micromonde algébrique d'ordre 1 incluront : (1) la représentation de nouveaux opérateurs, un opérateur de colonne pour représenter les systèmes d'équations dans la forme habituelle, \sum , \int , etc, (2) un glisser-déposer en mode équivalent plus puissant (3) la manipulation d'opérateurs, (4) la représentation et l'affichage de sous-problèmes, (5) l'extension de l'évaluation de l'équivalence et (6) une fonction « rejouer ». Bien sûr, nous ajouterons aussi des particularités et réaliserons des modifications suggérées par les expériences.

Nous envisageons de combiner ensuite ce micromonde avec nos travaux précédents, ajoutant des règles de transformations, des connaissances pour résoudre problèmes, pour évaluer les résolutions d'étudiants, pour expliquer, et pour fournir de l'aide.

7. Domaine de validité et discussion

Pour conclure, nous voudrions aborder le sujet du *domaine de validité* comme il a été présenté par Balacheff et Sutherland [94] qui ont défini quatre dimensions en mathématiques pour le *domaine de validité épistémologique*:

La première dimension est l'ensemble des problèmes que le micromonde permet de considérer. Pour l'activité de construction d'expressions dans le micromonde Aplusix, c'est n'importe quelle expression bien formée employant les opérateurs et les constantes inclus dans le système et respectant les types. Pour l'activité de raisonnement, l'ensemble des problèmes est constitué de n'importe quel problème concernant une expression algébrique qui appartient à la classe d'expressions sur lesquelles le système est capable de calculer l'équivalence. C'est actuellement les polynômes d'une variable avec n'importe quel type de forme et n'importe quel type de coefficients. Les équations avec à une inconnue et un degré inférieur à 5 seront incluses bientôt. D'autres classes d'expressions seront ajoutées plus tard. Noter que c'est indépendant du type

de problème (factoriser, développer, résoudre, etc). Le type de problème doit juste être un type de problème algébrique, c'est-à-dire, doit être tel qu'un problème soit invariant quand l'expression est remplacée par une expression équivalente.

La deuxième dimension est la nature des outils et les objets fournis par la structure formelle. Les objets du micromonde Aplusix sont des expressions algébriques telles qu'elles sont définies dans la théorie des réécritures [Dershowitz et Jouannaud 89], qui est la théorie la plus achevée sur les expressions. La nature de l'outil est de permettre de construire des expressions et de permettre de comparer ces expressions pour la relation d'équivalence.

La troisième dimension est la nature de la phénoménologie définie sur sa structure formelle. La phénoménologie à l'interface du micromonde Aplusix a un maximum de fidélité avec la représentation d'expressions. C'est notre principe premier et permanent pour l'interface : mettre en oeuvre *la représentation 2D habituelle*. Concernant les caractéristiques d'édition, comme cela a été exposé, nous les avons définies employant les concepts fondamentaux du domaine (la structure et la relation d'équivalence). L'affichage de l'équivalence est un phénomène naturel du fait que l'équivalence est une propriété fondamentale de tout raisonnement algébrique.

La quatrième dimension est le type de contrôle que le micromonde rend disponible aux utilisateurs et les retours fournis. Pour l'activité *de construction d'expressions*, l'élève peut exécuter n'importe quelle action d'édition. La différence principale avec d'autres éditeurs est que peu de malformation sont acceptées sur les expressions et que ces malformations sont indiquées (en bleu). Pour l'activité raisonnement, l'élève peut produire un nouveau pas à tout moment (seulement à partir d'une expression bien formée toutefois) et dispose de l'indication de l'équivalence comme rétroaction fondamentale. Cette sorte de rétroaction, déjà réalisé par McArthur [McArthur et al. 87] dans un contexte élémentaire, n'a pas été profondément expérimentée. Nous sommes impatients d'observer comment elle peut profiter à l'apprentissage des étudiants.

Balacheff et Sutherland [94] ont aussi évoqué un domaine didactique de validité pour les micromondes. Sur cette question, nous indiquerons juste deux points. D'abord, le micromonde Aplusix est fondamentalement un monde algébrique, et non pas un monde pré-algébrique. Par exemple, il ne comporte aucune soustraction. Le signe moins est un symbole unaire ($a-b$ est la somme de a et de l'opposé de b). Donc les élèves qui ne sont pas prêts à être plongés dans un contexte algébrique strict ne profiteront probablement pas de l'utilisation du système. Deuxièmement, le système peut être paramétré par l'utilisateur ou l'enseignant. Un fichier texte contient la description des opérateurs. Il est possible d'activer ou de désactiver un opérateur ; il est possible de changer certaines particularités d'un opérateur. Par exemple, on peut permettre ou non d'avoir des variables dans un dénominateur ou dans une racine carrée. Cela permet le paramétrage de l'environnement pour un niveau particulier.

Nous avons conçu un micromonde algébrique selon une voie épistémologique et nous finissons le premier prototype. Nous espérons que la plupart de nos choix seront bons, mais nous n'en sommes pas sûrs. La réponse viendra des utilisateurs et nous analyserons soigneusement les écarts entre le comportement du système et les espérances des utilisateurs, de manière à corriger des problèmes mineurs et apprendre des problèmes plus importants, que nous corrigerons aussi. Les problèmes importants et les leçons qu'ils fourniront, seront publiés.

8. References

[Adobe FrameMaker] V5.0, <http://www.adobe.com/products/framemaker/main.html>

[Balacheff & Sutherland 94] N. Balacheff, R. Sutherland, Epistemological domain of validity of microworlds, the case of Logo and Cabri-géomètre. In : Lewis R., Mendelshon P. (eds) Proceedings of the IFIP TC3/WG3.3 : Lessons from learning (pp.137-150). North-Holland, 1994.

[Beeson 96] M. Beeson, Design Principles of Mathpert: Software to support education in algebra and calculus, in: Kajler, N. (ed.) *Human Interfaces to Symbolic Computation*, Springer-Verlag, 1996.

[Bundy & Welham 81] A. Bundy and B. Welham. Using Meta-level Inference for Selective Application of Multiple Rewriting Rule Sets in Algebraic Manipulation. *Artificial Intelligence*, Vol 16, no 2, 1981.

[Dershowitz & Jouannaud 89] N. Dershowitz and J.P. Jouannaud. *Rewrite Systems*. In *Handbook of Theoretical Computer Science*, Vol B, Chap 15. North-Holland. 1989.

- [Foss 87] Foss C.L. *Learning from errors in ALGEBRALAND*. IRL report No IRL87-0003, 1987.
- [Graphing Calculator] V1.2, <http://www.PacificT.com>
- [Koedinger et al. 97] Koedinger K. R., Anderson J. R., Intelligent Tutoring Goes To School in the Big City. *International Journal of Artificial Intelligence in Education*, 8, 30-43. 1997.
- [Laborde 89] J.M. Laborde, Designing Intelligent Tutorial Systems: the case of geometry and Cabri-géomètre, *IFIP WG 3.1 Working Conference on Educational Software at the Secondary Education Level*, Reykjavik, 1989.
- [McArthur et al. 87] D. McArthur, C. Stasz and J. Y. Hotta. Learning problem-solving skills in algebra. *Journal of education technology systems*, 15, p 303-324, 1987.
- [Microsoft Equation] V3.0, <http://www.mathtype.com/msee>
- [Nguyen-Xuan et al. 93] A. Nguyen-Xuan, J.F. Nicaud, J.M. Gélis, F. Joly : An Experiment in Learning Algebra with an Intelligent Learning Environment. *Actes de PEG'93*, Edinbourg, 1993.
- [Nguyen-Xuan et al. 99] A. Nguyen-Xuan, A. Bastide, J.F. Nicaud, "Learning to match algebraic rules by solving problems and by studying examples with an intelligent learning environment, proceedings of *Artificial Intelligence in Education*, Le Mans, 1999.
- [Nicaud et al. 90] J.F. Nicaud, C. Aubertin, A. Nguyen-Xuan, M. Saïdi, P. Wach : APLUSIX: a learning environment for student acquisition of strategic knowledge in algebra. *Actes de PRICAI'90*. Nagoya, 1990.
- [Nicaud et al. 94] J.F. Nicaud, J.M. Gélis, M. Saïdi, A. Nguyen-Xuan, "The APLUSIX project: a computer-aided teaching of algebra. Methodology, theoretical foundations, realisations and experiments", *CALISCE*, p 369-376, Paris, septembre 1994.
- [Oliver & Zukerman 90] Oliver J., Zukerman I. *DISSOLVE: An Algebra Expert for an Intelligent Tutoring System*. Proceeding of ARCE, Tokyo, 1990.
- [Papert 80] S. Papert. *Mindstorms. Children Computers, and Powerful Ideas*. Sussex: Harvester.
- [StarOffice] V5.1 <http://www.sun.com/staroffice>
- [Thompson 87] P. W. Thompson, Mathematical Microworlds and Intelligent Computer Assisted Instruction, in *Artificial Intelligence and Instruction*, Addison Wesley, Reading MA-USA (1987).
- [Thompson 89] P. W. Thompson. Artificial Intelligence, Advanced Technology, and Learning and Teaching Algebra. In S. Wagner and C. Kieran: *Research issues in the learning and Teaching of Algebra*. Lawrence Erlbaum. 1989.